

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

```

;*****
;*
;*      Submit CCP/M-86
;*
;*****

;Revisions:
;      30 Mar 83  dh - attach after commands except chain
;      21 Mar 83  whf - cli on 'mov ss'; fix readchar eof
;      18 Feb 83  whf - %global/%local cmds
;      15 Feb 83  dh - closing windows

csseg

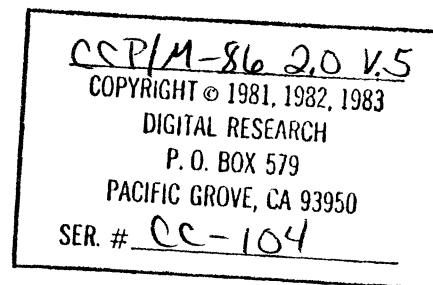
0000 EB38      003D      jmps submit

0002 5355424D4954      db      "SUBMIT 1.3 "
0012E3320
000D 434F50595249      db      "COPYRIGHT (C) 1983,"
0014F50595249
0018F50595249
001CF50595249
0020 204449474954      db      " DIGITAL RESEARCH "
002449474954
002849474954
0032 332F32312F38      db      "3/21/83"
003632312F38
0039 000A0024      db      13,10,0,"$"

00FF      true      equ      Offh
0000      false     equ      0
0000      unknown   equ      0
00E0      mpmint    equ      224      ; int vec for mpm ent.

0001      mpm_conin   equ      1
0002      mpm_conout  equ      2
0009      mpm_conwrite equ      9
000A      mpm_conread equ     10
000B      mpm_constat equ     11
000C      mpm_version equ     12
000E      mpm_diskselect equ     14
000F      mpm_openfile equ     15
0010      mpm_closefile equ     16
0014      mpm_readfile equ     20
0019      mpm_getdefdisk equ     25
001A      mpm_setdma  equ     26
0020      mpm_usercode equ     32
002F      mpm_chain   equ     47
008F      mpm_terminate equ    143
0091      mpm_setprior equ    145

```



```

54
55 0092 mpm_conattach equ 146
56 0093 mpm_condetach equ 147
57 0094 mpm_setdefcon equ 148
58 0096 mpm_clcmd equ 150
59 0098 mpm_parse equ 152
60 0099 mpm_getdefcon equ 153
61 009C mpm_getpdadr equ 156
62 00A0 mpm_setdeflst equ 160
63 00A4 mpm_getdeflst equ 164
64
65 ; fcb offsets
66 0006 mode2 equ 6 ; 2nd open mode byte
67 000A modesys equ 10 ; SYS attribute byte
68 0009 ftype equ 9 ; file type
69
70 004B sysdrive equ byte ptr .4bh ; system disk in SYSDAT
71
72 0003 e_no_memory equ 3 ; cant find memory
73 000C e_no_pd equ 12 ; no free pd's
74 000F e_q_full equ 15 ; full queue
75 0017 e_illdisk equ 23 ; illegal disk #
76 0018 e_badfname equ 24 ; illegal filename
77 0019 e_badftype equ 25 ; illegal filetype
78 001C e_bad_load equ 28 ; bad ret. from BDOS load
79 001D e_bad_read equ 29 ; bad ret. from BDOS read
80 001E e_bad_open equ 30 ; bad ret. from BDOS open
81 001F e_nullcmd equ 31 ; null command sent
82 0025 e_ill_lst equ 37 ; illegal list device
83 0026 e_ill_passwd equ 38 ; illegal password
84 0028 e_abort equ 40 ; aborted in CLI
85
86 0005 p_prior equ 05H ;process descriptor priority
87 0006 p_flag equ word ptr 06H ;flags
88 0008 p_name equ 08H ;name of process
89 001E p_parent equ 1EH ;PD's parent
90 0012 p_drv equ byte ptr 012H
91 0013 p_user equ byte ptr 013H
92 0024 p_list equ byte ptr 024H
93
94 0002 pf_keep equ 02H ;keep flag
95 0800 pf_childabort equ 800H ;child aborted abnormally
96 0080 pf_ctlc equ 080H ;control c occured
97
98 ;*****
99 ;*
100 ;* Submit Code
101 ;*
102 ;*****
103
104 ;=====
105 submit: ; PROGRAM MAIN - INITIALIZATION
106 ;=====

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

107
108 003D 8C08      mov ax,ds
109 003F 9C58FA    pushf 1 pop bx 1 cli      ;disable interrupts for stack switch
110 0042 8E00      mov ss,ax
111 0044 8C3C01    mov sp,offset stacktop
112 0047 5390      push bx 1 popf      ;restore interrupts
113
114 0049 B10C      mov cl,mpm_version
115 004B E85201    call mpm      01A0
116 004E 80E7FD    and bh,not 2
117 0051 80FF11740E 0064    cmp bh,11H 1 jz okvers      ;strip network bit
118 0056 80FF147409 0064    cmp bh,14H 1 jz okvers      ;check MP/N-86 or
119 005B 8A0307      mov dx,offset wrongvers      ;check Concurrent CP/M-86
120 005E E80E06      call print_string
121 0061 E9F405      jmp terminate      066F
122                                0658
123                                okvers:
124                                ;
125                                ;set priority better than parent
126                                ;
127
128 0064 B19C      mov cl,mpm_getpdadr
129 0066 E83701    call mpm      01A0      ;get our PD address
130 0069 891E4104    mov pdadr,bx      ;save PD address
131 006D 8C064304    mov sysdatseg,es      ;and the system data area segment
132 0071 26835F1E    mov bx,es:p_parent[bx] ;get our parent's PD address
133 0075 891E4504    mov parent_offset,bx ;save parent's PD addr
134 0079 83FA007505 0083    cmp bx,0 1 jne psetpar      ;see if parent exists
135 007E BAC500E800 0090    mov dx,197 1 jmps setp      ;oops, set to better than fmp
136 0083 26885705    psetpar:mov dx,es:p_prior[bx] ;our parent's priority
137 0087 4A          dec dx      ;we run at one better,
138 0088 83FA407303 0090    cmp dx,64 1 jae setp
139 008D BA4000      mov dx,64      ;don't compete w/system
140 0090 B191      setp:  mov cl,mpm_setprior
141 0092 E80801      call mpm      01A0      ;set our priority
142
143 0095 B11A      mov cl,mpm_setdma
144 0097 BA3C01      mov dx,offset dma      ;using default buffer at 80H for
145 009A E80301      call mpm      01A0      ;command tail and argument expansion
146
147 009D E8C605      call getuser      0666
148 00A0 A20A04      mov command_user,al
149 00A3 A20E04      mov defuser,al
150
151 00A6 BE5C00      mov si,offset fcb
152 00A9 C6440953    mov byte ptr ftype[si], 'S' ;look for file with type = SUB
153 00AD C6440A55    mov byte ptr ftype+1[si], 'U'
154 00B1 C6440B42    mov byte ptr ftype+2[si], 'B'
155 00B5 804C0680    or byte ptr mode2[si], 80h ;open in R/O mode
156 00B9 8B06      mov dx,si
157 00BB E8F805      call openfile      0686
158 00BE 3CFF      cmp al,0ffh
159 00C0 7532      jne exists      00F4

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

160
161
162 00C2 B200          mov dl,0          ;try user 0
163 00C4 88150A04      mov command_user,dl
164 00C8 E89D05        0568      call setuser
165 00CB 8E064304      mov es,sysdatses ;on system drive
166 00CF 26A04B00      mov al,es:sysdrive
167 00D3 FEC0          inc al
168 00D5 A25C00        mov fcb,al
169 00D8 BA5C00        mov dx,offset fcb
170 00DB E8D805        0696      call openfile
171 00DE 3CFF          cmp al,0ffh
172 00E0 7409          00EB      je openfail
173 00E2 BE5C00        mov si,offset fcb
174 00E5 F6440A80      test byte ptr modesys[si],080H ;does it have SYS attribute?
175 00E9 7509          00F4      jnz exists
176                                openfail:
177 00EB BA7308        mov dx,offset sopen_err
178 00EE E87E05        066F      call print_string
179 00F1 E94804        053C      jmp submitabort
180
181                                exists:
182 00F4 BA5C00        mov dx,offset fcb
183 00F7 E8C005        068A      call closefile
184 00FA A05C00        mov al,fcb          ;save disk number where command file is
185 00FD 3C00          cmp al,0          ;default disk ?
186 00FF 7505          0106      jnz save_disk_no
187 0101 E89A05        069E      call getdisk
188 0104 FEC0          inc al
189                                save_disk_no:
190 0106 A20B04        mov command_disk,al
191
192 0109 E88605        0692      call getconsole          ;save console number this
193 010C A20C04        mov command_con,al      ;program was run from
194
195
196                                ;
197                                ;deblank command tail
198                                ;
199 010F BE8100        mov si,offset cmdtail + 1
200 0112 BFFE01        mov di,offset deblankbuf
201                                firstwhites:
202 0115 803C20        cmp byte ptr [si]," "
203 0118 7502          011C      jnz tab
204 011A E805          0121      jmps morawhites
205                                tab:
206 011C 803C09        cmp byte ptr [si],9
207 011F 7503          0124      jnz deblank
208                                morewhites:
209 0121 46            inc si
210 0122 E8F1          0115      jmps firstwhites
211                                deblank:
212 0124 8A04          mov al,byte ptr [si]

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

213
214 0126 46          inc si
215 0127 3C20        cmp al," "
216 0129 7418        jz skipwhite
217 012b 3C09        cmp al,9          ;tab
218 012D 7414        jz skipwhite
219 012F EB00        jmps copychar
220
221          copychar:
222 0131 3C00        cmp al,0          ;end of of command tail ?
223 0133 7405        jz putlastblank
224 0135 8805        mov [di],al
225 0137 47          inc di
226 0138 EBE4        jmps deblank
227          putlastblank:
228 013A C60520       mov byte ptr [di]," "
229 013D 47          inc di
230 013E 8805        mov [di],al      ;ends with space and zero
231 0140 EB11        jmps dedone
232
233          skipmore:
234 0142 46          inc si
235          skipwhite:          ;input = si points at tab or blank
236 0143 8A04        mov al,[si]
237 0145 3C20        cmp al," "
238 0147 74F9        jz skipmore
239 0149 3C09        cmp al,9          ;tab
240 014B 74F5        jz skipmore
241 014D C60520       mov byte ptr [di]," "
242 0150 47          inc di
243 0151 EB01        jmps deblank      ;si points to non white char
244                                     ;di is next char to copy
245          dedone:
246
247          ;
248          ;fill argument tail
249          ;
250
251
252 0153 8DFE01       mov bp,offset deblankbuf
253 0156 BRBE01       mov bx,offset argtable
254 0159 33F6        xor si,si          ;pointer into deblankbuf
255 015B 8BFE        mov di,si          ;index into argtable
256          nxfill:
257 015D 8A02        mov al,byte ptr [bp + si]
258 015F 3C00        cmp al,0          ;0th argument is "<filename>.sub"
259 0161 7410        jz filldone
260 0163 8931        mov [bx + di],si
261 0165 47          inc di
262 0166 EB0200       call skiparg      ;skip over argument and blank
263 0169 EBF2        jmps nxfill
264
265          skiparg:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

266
267 016B 46 inc si
268 016C 803A20 cmp byte ptr [bp + si], ' '
269 016F 75F4 jnz skiparg 0160
270 0171 46 inc si ;char after blank
271 0172 C3 ret
272
273 filldone:
274 0173 893EBC01 mov hvargs,di
275 0177 E8F204 call crlf 066C
276
277 017A C605030380 mov cmdbuf,maxcmdlen ;get first command
278 017F BA0303 mov dx,offset cmdbuf
279 0182 E8C003 call readbuffer 0545
280 0185 83F800 cmp bx,0
281 0188 7511 jne submitexit 0193 ;EOF on first command
282
283 018A C605800280 mov lcmdbuf,maxcmdlen ;try next command
284 018F BA8002 mov dx,offset lcmdbuf
285 0192 E88003 call readbuffer 0545
286 0195 8B1E7F02 mov no_more_cmds,bl ;true if EOF
287 0199 EB08 jmps topofcmdloop 01A3 ;and look ahead to see
288 ;if there are more commands
289
290 ;=====
291 submitexit:
292 ;=====
293
294 019B B100BA0000 mov cl,0 ! mov dx,0
295 ;jmp mpm
296
297 ;===
298 mpm: ; INTERFACE ROUTINE FOR SYSTEM ENTRY POINTS
299 ;===
300
301 01A0 CDE0C3 int mpint ! ret
302
303 ;=====
304 topofcmdloop: ; LOOP FOREVER
305 ;=====
306
307 ;
308 ; print CR,LF if we just sent a command
309 ;
310
311 01A3 803E40040074 01B2 cmp cmdsent,false ! je noclearline
312 08 01B2
313 01AA C605400400 mov cmdsent,false
314 01AF E88A04 call crlf 066C
315
316 noclearline:
317
318 ;

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER # _____

```

319
320           ; set up and print user prompt
321           ;
322
323           ; get current default disk
324
325
326 01B2 EB8904      069E      call getdisk
327 01B5 A20D04      mov defdisk,al
328
329           ; get current default user #
330           ; this call should be made on every
331           ; loop in case the "USER" program
332           ; has changed the default user number
333
334 01B8 E84B04      0666      call getuser
335 01BB A20E04      mov defuser,al
336
337           ; create user part of prompt string
338
339 01BE C60611040D   mov promptutens,13
340 01C3 C60612040D   mov promptuones,13
341 01C8 84C0        test al,al                ;don't print user 0
342 01CA 7414        jz pprompt
343 01CC C606120430   mov promptuones,"0"
344 01D1 3C0A7207    010C      cmp al,10 ! jb ones
345 01D5 C606110431   mov promptutens,"1"
346 01DA 2C0A        sub al,10
347 01DC 00061204    ones:    add promptuones,al
348                        pprompt:
349           ; create disk part of prompt string
350
351 01E0 B04102060D04 mov al,"A" ! add al,defdisk
352 01E6 A21304      mov promptdisk,al
353
354           ; write user prompt
355
356 01E9 8A1004      mov dx,offset prompt
357 01EC 803E11040D75 01F6      cmp promptutens,13 ! jne writeprompt
358 03              01F6
359 01F3 BA1104      mov dx,offset promptutens
360 writeprompt:
361 01F6 E88004      0679      call print_ds_string
362
363           ;
364           ; copy command to cli buffer, make upper case and
365           ; expand command arguments
366           ;
367
368 01F9 BE0503      mov si,offset cmd                ;where command is
369 01FC BD8703      mov bp,offset clicmd            ;copy w/ expanded args here
370 01FF 33FF      xor di,di                          ;index into cli command
371 nextchar:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```

372
373      0201 8A04      mov al,[si]      ;char from command
374      0203 46        inc si
375      0204 3C24      cmp al,'$'      ;argument ?
376      0206 7407      jz argmaybe
377      020F 803C24      020F      noarg:
378      0208 E84F00      025A      call copyit
379      020B E3F4      0201      jcxz nextchar
380      020D E851      0270      jmps echo_cmd
381
382      argmaybe:
383      020F 803C24      cmp byte ptr [si],'$'      ;look for two '$' in a row
384      0212 7503      0217      jne numeric
385      0214 46        inc si
386      0215 EB1F      0208      jmps noarg
387      0217 803C30      numeric:
388      021A 72EC      0208      cmp byte ptr [si],'0'
389      021C 803C39      0208      jb noarg
390      021F 77E7      0208      cmp byte ptr [si],'9'
391      0221 33C0      0208      ja noarg
392      0223 8A04      realarg:
393      0225 2C30      xor ax,ax
394      0227 46        mov al,[si]
395      0228 803C30      sub al,'0'
396      022B 7211      023E      inc si
397      022D 803C39      023E      cmp byte ptr [si],'0'
398      0230 770C      023E      jb onedigit
399      0232 B20A      023E      cmp byte ptr [si],'9'
400      0234 F6E2      023E      ja onedigit
401      0236 8A14      mov dl,10
402      0238 46        mul dl
403      0239 80EA30      mov dl,[si]
404      023C 02C2      inc si
405      023E 8B08      sub dl,'0'
406      0240 7202      add al,dl
407      0242 8B89      onedigit:
408      0244 EB89      mov bx,ax
409      0246 803C39      copyarg:
410      0248 8A9FBE01      cmp bx,numargs
411      024B 7407      0248      jb oknum
412      024D 803C39      0201      jmps nextchar
413      024F 803C39      oknum:
414      0251 803C39      mov bl,argtable[bx]
415      0253 803C39      ;argument address rel to deblankbuf
416      0255 803C39      copymore:
417      0257 803C39      mov al,deblankbuf[bx]
418      0259 803C39      cmp al,' '
419      025B 803C39      0201      jz nextchar
420      025D 803C39      025A      call copyit
421      025F 803C39      024C      inc bx
422      0261 803C39      jmps copymore
423
424      copyit:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____


```

425
426      ;copy character in AL to location CP + DI, increment DI.
427      ;if AL = 0, or command buffer is full then return CX = 0ffffh
428      ;else return CX = 0H
429
430      025A 8803          mov [bp + di],al      ;mov to cli command buffer
431      025C 47           inc di
432      025D 33C9         xor cx,cx
433      025F 81FF8000     cmp di,maxcmdlen      ;cli buffer size
434      0263 7305         jae donecopy
435
436      0265 3C00         cmp al,0
437      0267 7401         jz donecopy
438      0269 C3           ret
439
440      026A 893E0804     donecopy: mov clicmdlen, di
441      026E 49           dec cx
442      026F C3           ret
443
444
445      0270 BE8703       echo_cmd: mov si, offset clicmd      ;print command line after
446      0273 880E0804     mov cx, clicmdlen      ;parameter substitution
447      0277 E86E03       call printcmdline
448
449      ;
450      ; echo newline
451      ;
452
453      027A E8EF03       call crlf
454
455      ;
456      ; make sure not a null command
457      ;
458
459      027D 8D1E0503     lea bx,cmd
460      0281 803E04030074 0304 cmp blen,0 ! je gonextcmd
461      7C               0304
462
463      0288 803F207405   0292 dblank: cmp byte ptr [bx],' ' ! je zapblank
464      028D 803F097507   0299 cmp byte ptr [bx],9 ! jne noblanks ;no tabs
465
466      0292 43FE0E0403EB 0288 zapblank: inc bx ! dec blen ! jmps dblank
467      EF               0288
468
469      0299 8D0605033BC3 02B2 noblanks: lea ax,cmd ! cmp ax,bx ! je chksemi
470      7411             02B2
471
472      02A1 1E0732ED8A0E      ; remove leading blanks
473      0403                push ds ! pop es ! xor ch,ch ! mov cl,blen
474      02A9 8BF808BF3FCF3    mov di,ax ! mov si,bx ! cld ! rep movsb
475      A4
476      02B0 8B08            mov bx,ax
477
478      chksemi:            ; see if line starts with semicolon

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

478
479 02B2 803F3B744D 0304      cmp byte ptr [bx],";" ! je gonextcmd
480
481
482 ; see if disk change - if "X:" change def disk to X
483 ;
484
485 02B7 803E04030275 0307      cmp olen,2 ! jne notdrive
486 49          0307
487 02BE 807F013A756D 0331      cmp byte ptr [bx],";" ! jne try_builtin
488
489          ; change default disk
490
491 02C4 8A17          mov dl,[bx]      ;get disk name
492 02C6 80E25F        and dl,5fh      ;make Upper Case
493 02C9 80EA41        sub dl,"A"      ;make disk number
494
495          ; check bounds
496
497 02CC 80FA007205 02D6      cmp dl,0 ! jb subex
498 02D1 80FA0F7612 02E8      cmp dl,15 ! jbe okdrive
499
500 02D6 8AD007E89303 066F      mov dx,offset errstr ! call print_string
501 02DC BA5A08E88D03 066F      mov dx,offset driverr ! call print_string !call crlf
502 E88703          066F
503 02E5 E95402          053C      jmp submitabort
504
505 02E8 52E8AE035A 069A      push dx ! call setdisk ! pop dx ; select default disk
506 02ED 803E470400      cmp fix_parent,false
507 02F2 7410          jz gonextcmd      ;don't change parent's drive
508 02F4 8B1E4504850B      mov bx,parent_offset ! test bx,bx
509 02FA 7408          jz gonextcmd      ;parent doesn't exist
510 02FC 8E064304          mov es,sysdatseg
511 0300 26885712          mov es:p_drv[bx],dl
512
513 0304 E9C901          04D0 gonextcmd:      jmp getnxtcmd
514
515 0307 BF8703          notdrive:      mov dl,offset clicmd
516 030A BE1C04          mov si, offset parseresult
517 030D E8E902          05F9      call parsefilename
518 0310 E311          0323      jcxz goodparse
519 0312 B17E          mov cl,126
520 0314 BF8703          mov di,offset clicmd
521 0317 B020          mov al," "
522 0319 0E07          push cs ! pop es
523 031B F2AE          repne scasb
524 031D C60524          mov byte ptr[di], "4"
525 0320 E90701          04FA      jmp clierror
526
527 0323 891E1A04          goodparse:      mov parseret,bx
528 0327 803E1C040074 0331      cmp parseresult,0 ! jz try_builtin
529 03          0331
530 032E E96D01          049E      jmp not_builtin

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

531
532
533          try_builtin:
534 0331 BF4804          mov di,offset usercmd
535 0334 BE1C0446        mov si,offset parseresult ! inc si
536 0338 1E07          push ds ! pop es
537 033A B90400F3A7      mov cx,4 ! repz cmpsw
538 033F 7565          jnz notuser
539
540 0341 BE1C04          mov si,offset parseresult
541 0344 8B3E1A0483FF    mov di,parseret ! cmp di,0
542 00
543 034B 7443          je pruser
544 034D E8A902          call parsefilename ; CX = 0 if ok
545 0350 83F900753B      cmp cx,0 ! jne pruser
546 0355 BE1C0446        mov si,offset parseresult ! inc si
547 0359 8B14          mov dx,[si]
548 035B E8AB02          call a_to_b
549 035E 80F80F7724      cmp bl,bl ! ja usererr
550 0363 8AD352          mov dl,bl ! push dx
551 0366 E8FF025A        call setuser ! pop dx
552 036A 8B150E04        mov defuser,dl
553 036E 803E470400      cmp fix_parent,false
554 0373 741B          jz pruser ;don't fix parent
555 0375 8B1E450485DB    mov bx,parent_offset ! test bx,bx
556 037B 7413          jz pruser ;parent doesn't exist
557 037D 8E064304        mov es,sydatseg
558 0381 26885713        mov esip_user[bx],dl
559 0385 E809          jmps pruser
560 0387 BA2207          mov dx,offset usererrmsg
561 038A E8E202          call print_string
562 038D E9AC01          jmp submitabort ;any error fatal
563 0390 BA3A07          pruser:
564 0393 E83902          mov dx,offset usermsg
565 0396 E8C002          call print_string
566 0399 8AD3E8A502      call getuser
567 039E E8C302E8C802    mov dl,bl ! call prnum
568 03A4 E92901          call crlf ! call crlf
569          notuser:      jmp getnxtcmd
570 03A7 BE1C0446        mov si,offset parseresult ! inc si
571 03AB BF5004          mov di,offset printercmd
572 03AE 1E07          push ds ! pop es
573 03B0 B90400F3A7      mov cx,4 ! repz cmpsw
574 03B5 7561          jnz notprinter
575
576 03B7 BE1C04          mov si,offset parseresult
577 03BA 8B3E1A0485FF    mov di,parseret ! test di,di
578 03C0 743F          jz prprinter
579 03C2 E83402          call parsefilename ; CX = 0 if ok
580 03C5 83F9007537      cmp cx,0 ! jne prprinter; no tail
581 03CA BE1C0446        mov si,offset parseresult ! inc si
582 03CE 8B14          mov dx,[si]
583 03D0 E83602          call a_to_b

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

584
585 0303 80F8FF7409 03E1      cmp dl,0ffh ! je printererr
586 0308 8AD352              mov dl,bl ! push dx
587 030B E8C4025A          06A2      call setlist ! pop dx
588 030F E309              03EA      jcxz setparent ;CX=0 if ok list #
589 03E1 BA4907              printererr: mov dx,offset printererrmsg
590 03E4 L88802E95201 066F      call print_string ! jmp submitabort ;any err fatal
591                          setparent:
592 03EA 803E470400          cmp fix_parent,false
593 03EF 7410              0401      jz prprinter ;don't fix parent
594 03F1 881E4504850B          mov bx,parent_offset ! test bx,bx
595 03F7 7408              0401      jz prprinter ;check for parent
596 03F9 8E054304              mov es,sydat_seg
597 03FD 26885724              mov es:p_list[bx],dl ;0 relative
598                          ;in CCP/M
599                          prprinter:
600 0401 BA6407              mov dx,offset printermsg
601 0404 E86802              066F      call print_string
602 0407 EB9C02              06A6      call getlist
603 040A 8AD3E83402          0643      mov dl,bl ! call prnum
604 040F E85A02E85702 066C      call crlf ! call crlf
605 0415 E98800              0400      jmp getnxtcmd
606
607                          notprinter:
608
609
610 0418 BE1C0446              mov si,offset parseresult ! inc si
611 041C BF5804              mov di,offset includcmd
612 041F 1E07              push ds ! pop es
613 0421 B90400F3A7          mov cx,4 ! repz cmpsw
614 0426 753A              0462      jnz notinclude
615
616 0428 BE1C04              mov si,offset parseresult
617 042B 8B3E1A0485FF          mov di,parseret ! test di,di
618 0431 742F              0462      jz notinclude
619 0433 E8C301              05F9      call parsefilename ; CX = 0 if ok
620 0436 83F900751E          0459      cmp cx,0 ! jne includerr; no tail
621 043B 8B364104              mov si,pdadr
622 043F 83C608              add si,p_name
623 0442 BF8703              mov di,offset clicmd
624 0445 B104              mov cl,4
625 0447 1E07              push ds ! pop es
626 0449 1E8E1E4304          push ds ! mov ds,sydatseg
627
628 044E F3A5              rep movsw ; overwrite $include with
629                          ; name of this program
630                          ; for recursive submits
630 0450 1F              pop ds
631 0451 BA8507              mov dx,offset includemsg
632 0454 E81802              066F      call print_string
633
634 0457 EB45              049E      jmps clicall ; send submit command
635
636 0459 BA7607              includerr: mov dx,offset includerrmsg

```

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

```

637
638 045C E81002      066F      call print_string
639 045F E9DA00      053C      jmp submitabort
640                                notinclude:
641                                ; check for "$global" option
642
643 0462 BE1C0446      mov si,offset parserresult 1 inc si
644 0466 BF6004      mov di,offset globalcmd
645 0469 1E07      push ds 1 pop es
646 046B B90400F3A7    mov cx,4 1 repz cmpsw
647 0470 750E      0490      jnz notglobcmd
648
649 0472 C6064704FF      mov fix_parent,true      ;for di, USER & PRINTER cmds
650 0477 BA9107      mov dx,offset globalmsg
651 047A E8F201      066F      call print_string
652 047D E95000      04D0      jmp getnxtcmd
653                                notglobcmd:
654                                ; check for "$local" option
655
656 0480 BE1C0446      mov si,offset parserresult 1 inc si
657 0484 BF6804      mov di,offset localcmd
658 0487 1E07      push ds 1 pop es
659 0489 B90400F3A7    mov cx,4 1 repz cmpsw
660 048E 750E      049E      jnz notlocal
661
662 0490 C606470400      mov fix_parent,false      ;for di, USER & PRINTER cmds
663 0495 BAA307      mov dx,offset localmsg
664 0498 E8D401      066F      call print_string
665 049B E93200      04D0      jmp getnxtcmd
666                                notlocal:
667
668                                notbuiltin:
669
670                                ;=====
671                                cli_call:      ; SEND CLI COMMAND
672                                ;=====
673
674                                ; Initialize CLI Control Block
675
676 049E C606860300      mov clicb_net,0
677                                ; make cli call
678
679 04A3 C6064004FF      mov cmdsent,true
680 04A8 803E7F02FF      cmp no_more_cmds,true
681 04AD 7513      04C2      jne gocli
682 04AF BA8703      mov dx,offset clicmd      ;this is the last command
683 04B2 B11A      mov cl,mpm_setdma
684 04B4 E8E9FC      01A0      call mpm
685 04B7 B12F      mov cl,mpm_chain
686 04B9 E8E4FC      01A0      call mpm
687                                ;returned from chain
688 04BC B91E00      mov cx,e_bad_open
689 04BF E93800      04FA      jmp clierror

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

690
691
692 04C2 8D158603B196      goto cli:      lea ix,click ! mov ci,mpm_click
693 04C8 E8D5FC          01A0      call mpm
694 04CB 83FB00752A      04FA      cmp dx,0 ! jne clierror
695                                ;japs get_nextcmd
696
697                                getnextcmd:
698 04D0 803E7F02FF          cmp no_more_cmds,true
699 04D5 7503          04DA      jne getnext2
700 04D7 E9C1FC          019B      jmp submitexit      ;terminate if no more commands
701                                getnext2:
702 04DA E8C901          06AA      call attach      ;get the console back
703                                ;before the TMP does
704
705 04DD BE8002          mov si,offset lcmbuf      ;copy look ahead command
706 04E0 8F0303          mov di,offset cndbuf      ;buffer to cndbuf
707 04E3 898300          mov cx,maxcmdlen+3      ;cmd buffer length
708 04E6 8CD8          mov ax,ds
709 04E8 8EC0          mov es,ax
710 04EA FC          cld
711 04EB F3A4          rep movsb
712
713 04ED 8AB002          mov dx,offset lcmbuf      ;get look ahead command
714 04F0 E85200          0545      call readbuffer
715 04F3 881E7F02          mov no_more_cmds,bl      ;true if EOF
716 04F7 E9A9FC          01A3      jmp topofcmdloop      ;and look ahead to see
717                                ;if there are more commands
718
719                                ;=====
720                                clierror:
721                                ;=====
722                                ; Cli call unsuccessful, analyze and display err msg
723                                ;      input: CX = ERROR CODE
724
725 04FA BEBF06          mov si,(offset clierrtab)-4
726 04FD 1E0E1F          push ds ! push cs ! pop ds
727                                nexterr:
728 0500 83C604          add si,4
729 0503 2E83BC0000FF 0512      cmp cs:word ptr [si],0ffffh ! je unknownerr
730                                7407      0512
731 0508 2E3B8C000075 0500      cmp cx,cs:[si] ! jne nexterr
732                                EE      0500
733                                unknownerr:
734 0512 2E8B940200          mov dx,cs:2[si]
735 0517 1F          pop ds
736                                ; jmps showerr
737
738
739                                showerr:      ; Print Error String
740                                ;-----      ; input: DX = string in CSEG
741
742 0518 52E85001      066C      perre: push dx ! call crlf

```

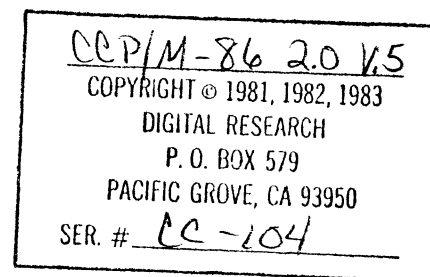
COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

743
744 051C BAD007E84001 066r      mov dx, offset errstr 1 call print_string
745 0522 5AE84901E843 066r      pop dx 1 call print_string 1 call crlf
746 01 066F
747 0529 BA6808E84001 066r      mov dx, offset cmdstr 1 call print_string
748 052F BE8703                mov si, offset clicmd
749 0532 880E0804                mov cx, clicmdlen
750 0536 E8AF09E83001 05E8      call printcmdline 1 call crlf
751                                ;jmp submitabort                    ;any error at this level is fatal
752
753                                ;=====
754 submitabort:
755                                ;=====
756                                ; tell user we are aborting
757 053C 6A6607E82D01 066F      mov dx, offset abortstr 1 call print_string
758 0542 E956FC                jmp submitexit
759
760                                ;=====
761 readbuffer:
762                                ;=====
763                                ; Simulate Read Console buffer from a disk file.
764                                ; input: DX = buffer address
765                                ; output: BX = 0ffffh if EOF
766
767 0545 86FA                mov di, dx
768 0547 32C0                xor al, al
769 0549 884501                mov [edi], al                ;no chars read
770 054C 3805                cmp [edi], al                ;zero length buffer
771 054E 7504                jnz buf_ok                ;return no error - nop
772 0550 8BF7FF                mov bx, 0ffffh                ;ready for error
773 0553 C3                ret
774
775 0554 8BF74646                buf_ok: mov si, di 1 inc si 1 inc si
776                                ;si is next char
777
778 read_another:
779 0558 8A4501                mov al, [edi]
780 055B 3A05                cmp al, [edi]
781 055D 731E                jae done_ok                ;full buffer
782 055F 57                push di
783 0560 E82700                call readchar                ;fills [si]
784 0563 5F                pop di
785 0564 83FB00                cmp bx, 0                ;non-zero or eof error
786 0567 7401                jz not_error
787 0569 C3                ret
788                                not_error:
789 056A 803C0A7502 0571      cmp byte ptr [si], 0ah 1 jne not_1f
790 056F E80C                jmps done_ok
791 0571 803C007407 057D not_1f: cmp byte ptr [si], 0dh 1 je done_ok
792
793 0576 FE4501                inc byte ptr [edi]                ;not cr, inc buffer count
794 0579 46                inc si                ;inc char position
795 057A E90BFF                jmp read_another
796                                done_ok:

```



```

796
797 057D 807D0100      cmp byte ptr [di],0      ;no characters read in buffer
798 0581 7405          jz read_another      ;skip leading cr's and lf's
799 0583 C60400      mov byte ptr [si],0
800 0586 BD0000C3      mov bx,0 ;ret
801
802                ;=====
803 readchar:          ;=====
804                ; Read a character from file.
805                ; input = SI is address of where to put character
806                ; output = BX = 0ffffh if error or EOF
807                ;
808
809 058A 32FF          xor bh,bh
810 058C 8A1E0F04      mov bl,buf_ptr      ;char ptr in dma buffer
811 0590 80F880      cmp bl,128
812 0593 7231          jb no_read
813 0595 56            push si
814 0596 8A160A04      mov dl,command_user    ;user number of command file
815 059A E8C800      call setuser
816 059D 8B5C00      mov bx,offset fcb
817 05A0 A00904      mov al,command_disk    ;disk drive number of command file
818 05A3 8807          mov [bx],al          ;re-open each time to allow deletion
819 05A5 804F0680      or byte ptr mode2Lbx],80h ;of command file, open in R/O mode
820 05A9 8BD3          mov dx,bx
821 05AB E80901      call openfile
822 05AE 3CFF          cmp al,0ffh
823 05B0 7428          jz readerr
824 05B2 8A5C00      mov dx,offset fcb
825 05B5 E80601      call readfile          ;read sequential
826 05B8 3C00          cmp al,0
827 05BA 751E          jnz readerr
828 05BC 8A5C00      mov dx,offset fcb
829 05BF E8F800      call closefile
830 05C2 8B0000      mov bx,0
831 05C5 5E            pop si
832
833 05C6 8A873C01      no_read:      mov al,dma[bx]
834 05CA 3C1A          cmp al,lah
835 05CC 7400          jz ret_eof      ;check for EOF
836 05CE 8804          mov [si],al
837 05D0 FEC3          inc bl
838 05D2 8B1E0F04      mov buf_ptr,bl    ;point to next char in disk buffer
839 05D6 3308          xor bx,bx
840 05D8 E804          jmps read_ret
841
842 readerr:
843 05DA 5E            pop si
844 ret_eof:
845 05DB 8BFFFF      mov bx,0ffffh    ;error or eof
846                ;jmps read_ret
847
848 read_ret:

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____


```

849
850 05DE 538A150E0418 0663      push bx ! mov dl,defuser ! call setuser ! pop ax
851      82005B      0663
852 05E7 C3                      ret
853
854      ;*****
855      ;*
856      ;*      SUBROUTINES
857      ;*
858      ;*****
859
860      printcmdline:      ;SI = string to print, CX = length
861 05E8 85C9      test cx,cx      ;print byte by byte
862 05EA 740C      05F8      jz blank      ;to include $ signs
863      moreecho:
864 05EC AC      lodsb
865 05ED 8AD0      mov dl,al
866 05EF 5156      push cx ! push si
867 05F1 E85E00      0662      call prchar
868 05F4 5E59      pop si ! pop cx
869 05F6 E2F4      05EC      loop moreecho
870      blank:
871 05F8 C3                      ret
872
873      parsefilename:      ;SI = fcb      DI = string
874 05F9 899800      mov cx,mpm_parse
875 05FC 8B1504      mov bx,offset pcb
876 05FF 893F897702      mov [bx],di ! mov 2[bx],si
877 0604 8B03E997F8      01A0      mov dx,bx ! jmp mpm
878
879      a_to_b:      ;dl = 1st char, dh = 2nd char
880 0609 80FE207504      0612      cmp dh," " ! jne atob2char
881 060E 8AF2B230      mov dh,dl ! mov dl,"0"
882 0612 80FE307229      0640      atob2char:      cmp dh,"0" ! jb atoberr
883 0617 80FE397724      0640      cmp dh,"9" ! ja atoberr
884 061C 80FA30721F      0640      cmp dl,"0" ! jb atoberr
885 0621 80FA39771A      0640      cmp dl,"9" ! ja atoberr
886 0626 80EE3080EA30      sub dh,"0" ! sub dl,"0"
887 062C B800008AC2      mov ax,0 ! mov al,dl
888 0631 52B10AF6E15A      push dx ! mov cl,10 ! mul cl ! pop dx
889 0637 8AD5B60003C2      mov dl,dh ! mov dh,0 ! add ax,dx
890 063D 89D8C3      mov bx,ax ! ret
891 0640 B3FFC3      atoberr:      mov bl,0ffh ! ret
892      prnum:      ; dl = num (0-15)
893 0643 80FA0A720A      0652      cmp dl,10 ! jb prnum_one
894 0648 52      push dx
895 0649 B231E81400      0662      mov dl,"1" ! call prchar
896 064E 5A80EA0A      pop dx ! sub dl,10
897 0652 80C230      prnum_one:      add dl,"0"
898 0655 E90A00      0662      jmp prchar
899
900 0658 33D28ACAEB62      06C0      terminate:      xor dx,dx ! mov cx,dx ! jmps mpm1
901 065E d101EB5E      06C0      charin:      mov cl,mpm_conin ! jmps mpm1

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. # _____

```

902
903 0662 B102EB5A 06C0 prechar: mov cl,mpm_conout 1 jmps mpm1
904 0666 B2FF getuser: mov dl,0ffh
905 0668 B120EB54 06C0 setuser: mov cl,mpm_usercode 1 jmps mpm1
906 066C BA1F07 crlf: mov dx,offset crlfstr
907 ;jmp printstring
908 066F 1E8CC88ED8 print_string: push ds 1 mov ax,cs 1 mov ds,ax
909 0674 E802001FC3 0679 call print_ds_string 1 pop ds 1 ret
910 print_ds_string:
911 0679 061E07 push es 1 push ds 1 pop es
912 067C 88F2 mov si,dx
913 067E 88FA mov di,dx
914 0680 89FFFF mov cx,0FFFFh
915 mask:
916 0683 AC lodsb
917 0684 247F and al,07FH ;mask parity
918 0686 AA stosb
919 0687 3C247402 068D cmp al,'1' 1 je print_it
920 068B E2F6 0693 loop mask
921 print_it:
922 068D 07 pop es
923 068E B109EB2E 06C0 mov cl,mpm_conwrite 1 jmps mpm1
924 0692 B199EB2A 06C0 getconsole: mov cl,mpm_getdefcon 1 jmps mpm1
925 0696 B194EB26 06C0 setconsole: mov cl,mpm_setdefcon 1 jmps mpm1
926 069A B10EEB22 06C0 setdisk: mov cl,mpm_diskselect 1 jmps mpm1
927 069E B119EB1E 06C0 getdisk: mov cl,mpm_getdefdisk 1 jmps mpm1
928 06A2 B1A0EB1A 06C0 setlist: mov cl,mpm_setdeflist 1 jmps mpm1
929 06A6 B1A4EB16 06C0 getlist: mov cl,mpm_getdeflist 1 jmps mpm1
930 06AA B192EB12 06C0 attach: mov cl,mpm_conattach 1 jmps mpm1
931 06AE B193EB0E 06C0 detach: mov cl,mpm_condetach 1 jmps mpm1
932 06B2 B10AEB0A 06C0 conread: mov cl,mpm_conread 1 jmps mpm1
933 06B6 B10FEB06 06C0 openfile: mov cl,mpm_openfile 1 jmps mpm1
934 06BA B110EB02 06C0 closefile: mov cl,mpm_closefile 1 jmps mpm1
935 06BE B114 readfile: mov cl,mpm_readfile 1; jmps mpm1
936 mpm1:
937 06C0 E900FA 01A0 jmp mpm
938
939
940 ;*****
941 ;*
942 ;* CODE DATA AREA - can be shared
943 ;*
944 ;*****
945
946 06C3 1F000000 clientab dw e_nullcmd, 0 ;null command
947 06C7 03000D07 dw e_no_memory, memerr ;No memory
948 06CB 0C00EF07 dw e_no_pd, pderr ;No unused PD
949 06CF 1800FD07 dw e_badfname, fnameerr;Ill. command
950 06D3 1700FD07 dw e_illdisk, fnameerr;Ill. disk
951 06D7 2600FD07 dw e_ill_passwd, fnameerr;Ill. password
952 06DB 1900FD07 dw e_badftype, fnameerr;Ill. type
953 06DF 1C001D08 dw e_bad_load, loaderr ;
954 06E3 10001D08 dw e_bad_read, loaderr ;

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

955
956 06E7 1E002808      dw  e_bad_open,      openerr ;
957 06E8 0F003808      dw  e_q_full,      qfullerr;
958 06EF 28005008      dw  e_abort,      aborterr;
959
960                      ; a few extra entries for future errors
961
962 06F3 FFFF0908      dw  0ffffh,      catcherr;
963 06F7 FFFF0808      dw  0ffffh,      catcherr;
964 06FB FFFF0P08      dw  0ffffh,      catcherr;
965 06FF FFFF0R08      dw  0ffffh,      catcherr;
966
967                      ;      MESSAGES
968
969 0703 526571756972   wrongvers      db  "Requires CCP/M-86 or MP/M-86"
970      657320434350
971      2F4D20383620
972      6F722040502F
973      402D3836
974 071F 0A0D24        crlfstr        db  10,13,"$"
975
976 0722 000A496E7651   usererrmsg     db  13,10,"Invalid User Number",13,10,"$"
977      6C6964205573
978      6572204E756D
979      625572000A24
980 073A 55736572204E   usermsg       db  "User Number = $"
981      756362657220
982      3D2024
983 0749 000A496E7661   printererrmsg db  13,10,"Invalid Printer Number",13,10,"$"
984      6C6964205072
985      696E74657220
986      4E756D626572
987      0D0A24
988 0764 5072696E7465   printermsg   db  "Printer Number = $"
989      72204E756D62
990      6572203D2024
991 0776 000A496E636C   includerrmsg db  13,10,"Include Errors$"
992      756465204572
993      726F7224
994 0786 287375626D69   includemsg   db  "(submit)",10,13,"$"
995      74290A0D24
996 0791 476C6F62616C   globalmsg    db  "Global mode ON",13,10,10,"$"
997      206D6F646520
998      4F4E0D0A0A24
999 07A3 476C6F62616C   localmsg     db  "Global mode OFF",13,10,10,"$"
1000      206D6F646520
1001      4F46460D0A0A
1002      24
1003 07B6 535542404954   abortstr     db  "SUBMIT can not continue",13,10,"$"
1004      2063616E206E
1005      6F7420636F6E
1006      74696E7565C0
1007      0A24

```

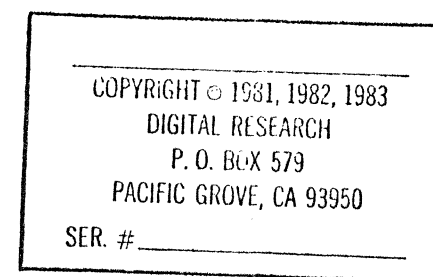
COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____



```

1008
1009
1010 07D0 43502F4D2045      errstr      db      "CP/M Error: 3"
1011      72726F723A20
1012      24
1013 07D0 4E6F7420456E      memerr      db      "Not Enough Memory!"
1014      6F756768204D
1015      656D6F727924
1016 07EF 504420546162      pderr      db      "PD Table Full!"
1017      6C652046756C
1018      6C24
1019 07F0 426164204669      fnameerr   db      "Bad File Spec!"
1020      6C6520537065
1021      6324
1022 080B 556E6B6E6F77      catcherr   db      "Unknown CLI Error!"      ;we should never print
1023      6E20434C4920
1024      4572726F7224
1025
1026
1027
1028
1029 081D 4C6F61642045      loaderr     db      "Load Error!"
1030      72726F7224
1031 082B 43616E277420      openerr     db      "Can't Find Command!"
1032      46696E642043
1033      6F6D6D616E64
1034      24
1035 083B 52535020436F      qfullerr    db      "RSP Command Que Full!"
1036      6D6D616E6420
1037      517565204675
1038      6C6C24
1039 0850 434C49204162      aborterr    db      "CLI Abort!"
1040      6F727424
1041 085A 496E76616C69      driverr     db      "Invalid Drive!"
1042      642044726976
1043      6524
1044 0868 436F6D6D616E      cmdstr      db      "Command = !"
1045      64203D2024
1046
1047 0873 4E6F20275355      sopen_err    db      "No ""SUB"" File Present",10,13, "!"
1048      42272046696C
1049      652050726573
1050      656E740A0D24
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060

```

```

;*****
;*
;*      Data Area - separate per invocation of SUBMIT
;*
;*****

      dseg
      org      05cH

```

```

1061
1062
1063      005C      rcb      rb      32
1064      007C 00      cr      db      0      ;current record
1065      007D      r0      rw      0
1066      007D      r2      rb      0
1067
1068      org      080h
1069
1070      0080      cmdtail      rb      128
1071
1072      0100 CCCCCCCCCCCC      stack      dw      0ccccch,0ccccch,0ccccch
1073      0106 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1074      010C CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1075      0112 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1076      0118 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1077      011E CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1078      0124 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1079      012A CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1080      0130 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1081      0136 CCCCCCCCCCCC      dw      0ccccch,0ccccch,0ccccch
1082      013C      stacktop      rw      0
1083
1084      0080      maxcmdlen      equ      128
1085
1086      ; the Read Console buffer and the
1087      ; C11 Control Block share the same memory
1088
1089      013C      dma      rb      128
1090
1091      018C      numargs      rw      1
1092      018E      argtable      rb      64      ;address of arguments in cmdtail
1093      01FE      deblankbuf      rb      maxcmdlen + 1
1094
1095      027F 00      no_more_cmds      db      false      ;set to true when EOF is
1096      ;found
1097      0280      lcmdbuf      rb      1
1098      0281      lflen      rb      1      ;read from disk to here,
1099      0282      lcmd      rb      maxcmdlen + 1      ;copy this to CMD
1100
1101      0303      cmdbuf      rb      1      ;current cmd goes here
1102      0304      blen      rb      1
1103      0305      cmd      rb      maxcmdlen + 1      ;copy w/ expanded args to C11CMD
1104
1105      0386      clicb      rb      0      ;cmd w/ expanded args goes here
1106      0386 00      clicb_net      db      0
1107      0387      clicmd      rb      maxcmdlen+1
1108      0408 0000      clicmdlen      dw      0
1109
1110      040A      command_user      rb      1
1111      040B      command_disk      rb      1
1112      040C      command_con      rb      1
1113      040D      defdisk      rb      1

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```

1114
1115 040E      defuser      rb      1
1116 040F 80    buf_ptr     db      120      ;force initial read
1117                                     ;in readchar routine
1118 0410 00      prompt     db      13
1119 0411 30      promptuns   db      "0"
1120 0412 30      promptunes  db      "0"
1121 0413 41      promptdisk  db      "A"
1122 0414 3E24    promptend   db      ">"
1123
1124 0416 8703    pcb         dw      offset clicmd
1125 0418 1C04                dw      offset parseresult
1126
1127 041A 0000    parseret     dw      0
1128 041C        parseresult  rb      36
1129 0440 00      cmdsent     db      false
1130
1131 0441        pdadr         rw      1      ;our pd address
1132 0443        sysdatseg     rw      1      ;the system data area's segment
1133 0445        parent_offset rw      1      ;parents pd address
1134 0447 00      fix_parent   db      false ;local/global mode
1135
1136 0448 555345522020 usercmd  db      "USER  "
1137 2020
1138 0450 5052494E5445 printerdb  db      "PRINTER "
1139 5220
1140 0458 24494E434C55 includecmd db      "$INCLUDE"
1141 4445
1142 0460 24474C4F4241 globalcmd  db      "$GLOBAL "
1143 4C20
1144 0468 244C4F43414C localcmd   db      "$LOCAL  "
1145 2020
1146
1147      end
1148
1149
1150 END OF ASSEMBLY.  NUMBER OF ERRORS: 0.  USE FACTOR: 17%

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ABORTERR	0850 V	958	1039#																
ABURISIR	0786 V	757	1003#																
ARGMAYBE	020F L	376	332#																
ARGTABLE	01BE V	253	414	1092#															
ATOB	0609 L	548	533	679#															
ATOBZCHAR	0612 L	880	882#																
ATOBERR	0640 L	882	883	884	885	891#													
ATTACH	06AA L	702	930#																
BLANK	05F8 L	862	870#																
BLEN	0304 V	469	466	472	485	1102#													
BUFOK	0554 L	771	774#																
BUFPTR	040F V	810	838	1116#															
CATCHERR	080B V	962	963	964	965	1022#													
CHARIN	065E L	901#																	
CHKSEMI	02B2 L	469	477#																
CLICALL	049E L	634	671#																
CLICB	0386 V	692	1105#																
CLICENET	0386 V	676	1106#																
CLICMD	0387 V	369	445	515	520	623	682	748	1107#	1124									
CLICMDLEN	0408 V	440	446	749	1108#														
CLIERROR	04FA L	525	689	694	720#														
CLIERRIAB	06C3 V	725	946#																
CLOSEFILE	068A L	183	829	934#															
CMD	0305 V	368	459	469	1103#														
CMDRBUF	0303 V	277	278	706	1101#														
CMDSENT	0440 V	311	313	679	1129#														
CMDSTR	0868 V	747	1044#																
CMDTAIL	0080 V	199	1070#																
COMMANDCON	040C V	193	1112#																
COMMANDDISK	040B V	190	817	1111#															
COMMANDUSER	040A V	148	163	814	1110#														
CONREAD	06B2 L	932#																	
COPYARG	0240 L	409#																	
COPYCHAR	0131 L	219	221#																
COPYIT	025A L	378	419	424#															
COPYMOKE	024C L	415#	421																
CR	007C V	1064#																	
CRLF	066C L	275	314	453	501	567	567	604	604	742	745								
		750	906#																
CRLFSTR	071F V	906	974#																
CS	SREG V	522	726	729	731	734	908												
DBLANK	0288 L	462#	466																
DEBLANK	0124 L	207	211#	226	243														
DEBLANKBUF	01FE V	200	252	416	1093#														
DEBONE	0153 L	231	245#																
DEFDISK	040D V	327	351	1113#															
DEFUSER	040E V	149	335	552	850	1115#													
DETACH	06AE L	931#																	
DMA	013C V	144	833	1089#															
DUNECOPY	026A L	434	437	439#															
DONEOK	057D L	780	789	790	795#														
DRVERR	085A V	501	1041#																
DS	SREG V	108	472	536	572	612	625	626	626	630	645								

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

EAUORT	0028 N	658	708	726	726	735	903	908	909	911
EBAOFNAME	0018 N	94#	953							
EBAOFIYPE	0019 N	76#	949							
EBAULOAD	001C N	78#	953							
EBAADOPEN	001E N	80#	688	956						
EBAADREAD	001D N	79#	954							
ELCICMD	0270 L	380	444#							
EILLDISK	0017 N	75#	950							
EILLISI	0025 N	82#								
EILLPASSWD	0026 N	83#	951							
ENUMEMORY	0003 N	72#	947							
ENUPD	000C N	73#	948							
ENULLCMD	001F N	81#	946							
EQFULL	000F N	74#	957							
ERRSIR	0700 V	500	744	1010#						
ES	SREG V	131	132	136	165	166	472	510	511	522
		557	558	572	596	597	612	625	645	656
		911	911	922						709
EXISTS	00F4 L	159	175	181#						
FALSE	0000 N	34#	311	313	506	553	592	662	1095	1129
FCB	005C V	151	168	169	173	182	184	816	824	828
FILLDONE	0173 L	259	273#							1063#
FIRSTWHILES	0115 L	201#	210							
FIXPARENT	0447 V	506	553	592	649	662	1134#			
FNAMEERR	07FD V	949	950	951	952	1019#				
FTYPE	0009 N	68#	152	153	154					
GEICUNSOLE	0692 L	192	924#							
GEIDISK	069E L	187	326	927#						
GEILIS1	06A6 L	602	929#							
GEINXT2	040A L	699	701#							
GEINXTCMD	0400 L	513	568	605	652	665	697#			
GEIUSER	0666 L	147	334	565	904#					
GLOBALCMD	0460 V	644	1142#							
GLOBALMSG	0791 V	650	936#							
GOCLI	04C2 L	681	691#							
GONEXTCMD	0304 L	460	479	507	509	513#				
GOODPARSE	0323 L	518	527#							
INCLUDECMD	0458 V	611	1140#							
INCLUDEMSG	0786 V	631	994#							
INCLUDEERR	0459 L	620	636#							
INCLUDERRMSG	0776 V	636	991#							
LBLEN	0281 V	1098#								
LCMD	0282 V	1099#								
LCMDBUF	0280 V	283	284	705	713	1097#				
LOADERR	031D V	953	954	1029#						
LOCALCMD	0468 V	657	1144#							
LOCALMSG	07A3 V	663	999#							
MASK	0683 L	915#	920							
MAXCMDLEN	0080 N	277	283	433	707	1084#	1093	1099	1103	1107
MEMERR	07DD V	947	1013#							
MODE2	0006 N	66#	155	819						
MODESYS	000A N	67#	174							

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

MOREECHO	05EC L	865#	869								
MOREWHITES	0121 L	204	208#								
MPM	0150 L	115	129	141	145	253#	624	686	693	877	937
MPMI	06C0 L	900	901	903	905	923	924	925	926	927	928
		929	930	931	932	933	934	935#			
MPMCHAIN	002F N	51#	685								
MPMCLICMD	0096 N	58#	692								
MPMCLUSEFILE	0010 N	46#	934								
MPMCUNATTACH	0092 N	55#	930								
MPMCUNDETACH	0093 N	56#	931								
MPMCUNIN	0001 N	38#	901								
MPMCUNOUT	0002 N	39#	903								
MPMCUNREAD	000A N	41#	932								
MPMCUNSTAT	000B N	42#									
MPMCUNWRITE	0009 N	46#	923								
MPMDISKSELECT	000E N	44#	926								
MPMGETDEFCON	0099 N	60#	924								
MPMGETDEFDISK	0019 N	48#	927								
MPMGETDEFDST	00A4 N	63#	929								
MPMGETPDADR	009C N	61#	128								
MPMLNT	00E0 N	36#	301								
MPMOPENFILE	000F N	45#	933								
MPMPARSE	0098 N	59#	874								
MPMREADFILE	0014 N	47#	935								
MPMSETDEFCON	0094 N	57#	925								
MPMSETDEFDST	00A0 N	62#	928								
MPMSETDMA	001A N	49#	143	683							
MPMSETPRIOR	0091 N	53#	140								
MPMTERMINATE	008F N	52#									
MPMUSERCODE	0020 N	50#	905								
MPMVERSION	000C N	43#	114								
NEXTCHAR	0201 L	371#	379	412	418						
NEXTERR	0500 L	727#	731								
NOARG	0208 L	377#	386	389	391						
NOBLANKS	0299 L	464	468#								
NOCLEARLINE	01B2 L	311	316#								
NOMORECMDS	027F V	286	680	698	715	1095#					
NOREAD	05C6 L	812	832#								
NOIBUILTIN	049E L	530	668#								
NOIDRIVE	0307 L	485	515#								
NOIERROR	056A L	785	787#								
NOIGLUBCMD	0480 L	647	653#								
NOIINCLUDE	0462 L	614	618	640#							
NOILF	0571 L	788	790#								
NOILOCAL	049E L	660	666#								
NOIPRINTER	0418 L	574	607#								
NOIUSER	03A7 L	538	569#								
NUMARGS	01BC V	274	410	1091#							
NUMERIC	0217 L	384	387#								
NXIFILL	015D L	256#	263								
OKDRIVE	0258 L	498	504#								
OKNUM	0248 L	411	413#								
OKVERS	0064 L	117	118	122#							

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

ONEDIGIT	023E L	398	400	407#															
ONES	010C L	344	347#																
OPENERR	0328 V	956	1031#																
OPENFAIL	00EB L	172	176#																
OPENFILE	06B6 L	157	170	821	933#														
PARENTOFFSET	0445 V	133	508	555	594	1133#													
PARSEFILENAME	05F9 L	517	544	579	619	373#													
PARSERESULT	041C V	516	528	535	540	546	576	576	581	610	616								
		643	656	1125	1128#														
PARSERET	041A V	527	541	577	617	1127#													
PCB	0416 V	875	1124#																
PDADR	0441 V	130	621	1131#															
PDERR	07EF V	948	1016#																
PDRV	0012 N	90#	511																
PERR	0518 L	742#																	
PCHILDABORT	0800 N	95#																	
PFCILC	0080 N	96#																	
PFKEEP	0002 N	94#																	
PFLAG	0006 N	87#																	
PLIST	0024 N	92#	597																
PNAME	0008 N	88#	622																
PPARENT	001E N	89#	132																
PPRIOR	0005 N	86#	136																
PRCHAR	0662 L	867	895	898	903#														
PRINICHOINE	05E8 L	447	750	860#															
PRINTOSSSTRING	0679 L	361	909	910#															
PRINTERCMD	0450 V	571	1138#																
PRINTERERR	03E1 L	585	589#																
PRINTERERRMSG	0749 V	589	983#																
PRINTERMSG	0764 V	600	988#																
PRINIT	068D L	919	921#																
PRINISRING	066F L	120	178	500	501	561	564	590	601	632	638								
		651	664	744	745	747	757	908#											
PRNUM	0643 L	566	603	392#															
PRNUMONE	0652 L	893	897#																
PROMPT	0410 V	356	1118#																
PROMPTDISK	0413 V	352	1121#																
PROMPTEND	0414 V	1122#																	
PROMPTUONES	0412 V	340	343	347	1120#														
PROMPTUTENS	0411 V	339	345	357	359	1119#													
PRPRINTER	0401 L	578	580	593	595	599#													
PRPROMPT	01E0 L	342	348#																
PRUSER	0390 L	543	545	554	556	559	563#												
PSETPAR	0083 L	134	136#																
PUSER	0013 N	91#	558																
PUIIASIBLANK	013A L	223	227#																
QFULLERR	083B V	957	1035#																
R0	007D V	1065#																	
R2	007D V	1066#																	
READANOTHER	0558 L	777#	794	798															
READBUFFER	0545 L	279	285	714	761#														
READCHAR	058A L	782	803#																
READERR	05DA L	823	827	842#															

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

READFILE	06RE L	825	937				
READRET	05DE L	846	843#				
REALARG	0221 L	392#					
RELEOF	05DB L	835	844#				
SAVEDISKNO	0106 L	186	199#				
SEICONSOLE	0696 L	925#					
SEIDISK	069A L	505	926#				
SEILIST	06A2 L	587	928#				
SEIP	0090 L	135	138	140#			
SEIPARENT	03EA L	588	591#				
SEIUSER	0668 L	164	551	815	850	905#	
SHOWERR	0518 L	739#					
SKIPARG	016B L	252	255#	269			
SKIPMORE	0142 L	233#	238	240			
SKIPWHITE	0143 L	216	213	235#			
SUPERRR	0873 V	177	1047#				
SS	SREG V	110					
STACK	0100 V	1072#					
STACKTOP	013C V	111	1082#				
SUBEX	0206 L	497	499#				
SUBMIT	0030 L	16	105#				
SUBMITABORT	053C L	179	503	562	590	639	754#
SUBMITEXIT	019B L	281	291#	700	758		
SYSDATSEG	0443 V	131	165	510	557	596	626 1132#
SYSORIVE	0043 V	70#	166				
TAB	011C L	203	205#				
TERMINATE	0658 L	121	900#				
TESTEND	0265 L	435#					
TOPOFFCNLOOP	01A3 L	287	304#	716			
TRUE	00FF N	33#	649	679	680	698	
TRYBUILTIN	0331 L	497	528	533#			
UNKNOWN	0000 N	35#					
UNKNOWNERR	0512 L	729	733#				
USERCMD	0448 V	534	1136#				
USERERR	0387 L	549	560#				
USERERRMSG	0722 V	560	976#				
USERMSG	073A V	565	980#				
WRITEPROMPT	01F6 L	357	360#				
WRONGVERS	0703 V	119	969#				
ZAPBLANK	0292 L	463	465#				

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

